GRAMEX: Generating Controlled Grammar Exercises from Various Sources

Guillaume Toussaint[1]
(1) CNRS, LORIA
Vandœuvre-Lès-Nancy
France

Yannick Parmentier[2,3]
(2) Université de Lorraine, LORIA
(3) Université d'Orléans, LIFO
Vandœuvre-Lès-Nancy, France

Claire Gardent[1]
(1) CNRS, LORIA
Vandœuvre-Lès-Nancy
France

firstname.lastname@loria.fr

Abstract

This paper presents GRAMEX, an application designed to assist teachers in the creation of learning materials, namely grammar exercises. More precisely GRAMEX leverages state-of-the-art parsing techniques to morpho-syntactically annotate texts and turn these into grammar exercises while aligning these with official curricula. Allowing teachers to freely select excerpts of texts from which to generate specific grammar exercises aims to increase learners' engagement in educational activities. GRAMEX currently supports 4 types of exercises (Fill-in-the-Blanks, Mark-the-Words, Single and Mutliple Choice questionnaires) and 3 output formats (JSON objects, printable workbooks, H5P interactive content). GRAMEX is under active development and has been experimentally used with teachers of L1-learners in elementary and middle French schools.

1 Introduction

Grammar learning is known to have a strong impact on language learning in general. Indeed, studies showed that a lack of self-confidence in one's own grammatical skills often leads to broader difficulties in language learning and writing (Ignacia-Dorronzoro and Klett, 2007; Castagné-Véziès, 2018). Further investigations also suggest that isolating grammar practice from other learning activities results in higher learning difficulties (Vincent, 2016). This, combined with the positive effects on learners' motivation observed by Peacock (1997), advocates for the use of authentic texts (possibly seen in various contexts) as a valuable resource for automatic generation of grammar exercises.

The GRAMEX project builds on this idea to provide (1) teachers with a digital environment which can be used to generate grammar exercises from user-defined texts and learning goals and (2) learners with an online facility to train and monitor their progress. The generated exercises are annotated with fine-grained morphological and syntactic information along with readability scores (François and Fairon, 2012) and links to official curricula, allowing teachers to control exercise generation, ensuring the adequacy of the output material for target learners.

Along with this **control** on exercise generation, GRAMEX features include:

robustness : the use of efficient neural parsing techniques combined with error analysis on parse trees makes it possible to filter out sentences leading to ill-formed questions ;

multilingualism : two languages have been tested so far (French and English), yet GRAMEX relies on multilingual parsing engines covering 20+ languages ;

extensibility : thanks to its modular architecture, GRAMEX can easily be extended to other languages or new exercise types (see Section 3) ;

interoperability : GRAMEX comes with a REST Application Programming Interface (API) and 3 export formats (JSON objects, printable workbooks in docx format, interactive content in H5P format), allowing users to interact with GRAMEX in many ways, including within Learning Management Systems (LMS), external applications or in a classical paper-based setting.

The remaining of this paper is organized as follows. In Section 2, we present related work. In Section 3, we describe how teachers can use GRAMEX to generate grammar exercises, and how learners can complete these. In Section 4, we present GRAMEX's implementation. In Section 5,

Guillaume Toussaint, Yannick Parmentier and Claire Gardent. GRAMEX: Generating Controlled Grammar Exercises from Various Sources. *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2024)*. Linköping Electronic Conference Proceedings 211: 226–234.

we comment on GRAMEX status and discuss its current limitations and ongoing work. We finally conclude and present future work in Section 6.

2 Related work

There have been many approaches to automatic generation of grammar exercises over the last decades. The corresponding systems distinguish themselves according to their core functionalities.

A first distinction can be made between systems supporting **custom text input** and those relying on predefined resources (corpora, grammars and / or lexicons). The latter includes *ArikIturri* (Aldabe et al., 2006), *Grammar Exerciser* (Perez-Beltrachini et al., 2012) and *Lärka* (Volodina et al., 2014). Systems allowing users to enter free text, like GRAMEX, include *MIRTO* (Antoniadis et al., 2006), *Sakumon* (Hoshino and Nakagawa, 2007), *VIEW* (Meurers et al., 2010), *Language Muse* (Madnani et al., 2016), *Language Exercise App* (Perez and Cuadros, 2017), and *FLAIR* (Heck and Meurers, 2022b). These notably differ in the way input texts are pre-processed to extract candidate sentences. In the case of GRAMEX, sentence filtering is done by means of fine-grained morpho-syntactic annotations computed by state-of-the-art text parsers (namely, SpaCy[1] (Honnibal and Johnson, 2015) and Stanza[2] (Qi et al., 2020))[3] combined in tailored NLP pipelines (see Section 3), while other systems rely either on partial analyses involving specific part-of-speech tags or syntactic patterns (e.g. *MIRTO*, *Lärka*, *Sakumon*),[4] or on more abstract representations such as sentence vectorization (e.g. *Language Exercise App*).

A second distinction concerns their **degree of automation**. Most systems require human intervention (i.e., post-edition of questions, such as the selection of distractors in Multiple Choice questions) to create ready-to-use grammar exercises. GRAMEX is designed to limit such intervention as much as possible. Users are merely required to validate (and optionally reorder) output questions. This design choice is questionable, and may be revised in the light of experimental studies involving school teachers to be carried out in a near future.

A third distinction can be made on the **level of control** offered by these systems. Systems generally offer a limited control on the generation of exercises. Noticeable exceptions include *Language Exercise App*, where users can define target constructions, *MIRTO*, where users can also link questions to references providing learners with helpful information, *Language Muse*, which generates about 24 predefined activities at various levels (sentence, paragraph, discourse) and *FLAIR*, which comes with a highly configurable generation process, where users can for instance define additional parameters depending on the target grammatical phenomenon (Heck and Meurers, 2022b). In our case, a trade-off between configurability and usability is being sought. GRAMEX currently allows users to target precise predefined grammatical concepts extracted from official curricula (MENJS, 2018). A more fine-grained control is under development, allowing for instance to select target syntactic structures (see Section 5).

A fourth distinction concerns their **expressivity**, that is, the types of exercises they support. Most systems support Multiple Choice (MC) questionnaires since these can be automatically processed to evaluate learners' performances. The number and types of supported exercises vary from one system to another. GRAMEX currently supports 4 exercise types, namely Fill-in-the-Blanks (FiB), Mark-the-Words (MtW), Multiple Choice (MC) and Single Choice (SC). Other common exercise types, not yet supported by GRAMEX include Error Detection (ED), Memory (Mem), Shuffle (Sh) and Word Forms (WF). Table 1 summarizes the expressivity of the above-mentioned systems with respect to these types.

Finally, let us note that relatively few systems are able to **export** exercises to be integrated in external tools (i.e., Learning Management Systems) out-of-the-box.[5] Such systems include *VIEW*, which is a browser extension and as such can be integrated natively with web interfaces, *Language Exercise App* and GRAMEX, which can both export exercises in H5P format (interactive HTML5 content)[6] supported by many LMS.

[1] https://spacy.io

[2] https://github.com/stanfordnlp/stanza

[3] Note that these are not limited to syntactic analysis sensu stricto, they include many (neural and / or symbolic) modules for broader text analysis.

[4] Like GRAMEX, FLAIR uses state-of-the-art parsers, but only specific annotations are considered for exercise generation, following work of Pilán et al. (2016) on candidate sentence selection.

[5] *ArikIturri* exports exercises in XML format, which is not directly usable e.g. in an LMS, but can be relatively easily converted to other formats for integration.

[6] https://h5p.org/

System	SC	MC	ED	MtW	FiB	Mem	Sh	WF	Other
MIRTO				×	×				
ArikIturri		×	×		×			×	
FAST		×	×						
Sakumon		×							
VIEW		×		×	×				
Grammar Exerciser				×		×			
Lärka		×							
Language Muse	×	×							×
Language Exercise App		×		×	×		×		
FLAIR		×		×	×	×	×	×	
GRAMEX	×	×		×	×				

Table 1: Exercise types supported by exercise generation systems (these are in chronological order)

3 Workflow description

In a nutshell, GRAMEX is a web application allowing teachers to create exercises from custom texts depending on target grammatical phenomena and learner levels. These exercises can be shared with other users or exported for reuse in other applications (e.g. LMS). Teachers can furthermore create collections of activities (so-called lessons) which follow given learning paths. In the following subsections, we go through the various steps involved in exercise generation.

3.1 Selecting and annotating input data

In order to generate exercises, users need to first select an adequate input text.[7] They can select from the following sources : Wikipedia articles, web pages (identified by their URL), local files[8] and free (e.g. copy-pasted) texts.

From this source, the text is extracted (i.e., formatting information is removed) and fed to a custom yet classical NLP pipeline for text annotation. This pipeline builds on state-of-the-art parsers to performs various tasks sequentially: sentence segmentation, tokenization, part-of-speech tagging, morphological analysis and syntactic dependency parsing. As a result, each sentence from the input text is annotated with morpho-syntactic information in CoNLL format (Buchholz and Marsi, 2006) and stored in GRAMEX's database.

Additionally, we also compute and store, for each annotated sentence, its readability scores (e.g. Flesch–Kincaid (Kincaid et al., 1975)).[9] These scores may be used by teachers to order exercises depending on their readability or to adapt activities to pupils with special educational needs.[10]

The sentences which have been annotated (e.g., whose length is above a given threshold and which contain at least one of the target grammatical phenomena) can be inspected as illustrated in Figure 1. Sentences which should not be used in exercises (e.g. due to an inadequate vocabulary) can be manually filtered out by teachers at this step.

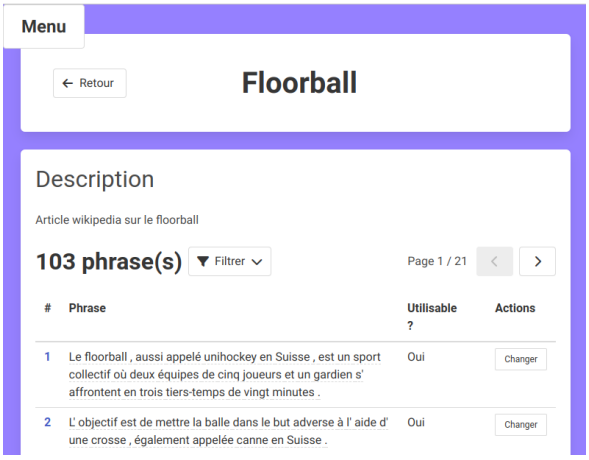

Figure 1: Teacher's interface to inspect annotated texts

[7] In our approach, the adequacy between a text and a target grammatical phenomenon is *by design* left to the teacher.

[8] For now, only text and pdf files are allowed, docx files will be supported soon.

[9] Other readability assessment Machine Learning-based techniques have been implemented following work by Hernandez et al. (2022), see (Ngo and Parmentier, 2023) ; their precision on representative data is yet to be evaluated prior to integrating these into GRAMEX.

[10] Currently GRAMEX does not use these readability scores for exercise generation, they are only displayed to teachers in order to help them to select which sentences to use in exercises. An automatic ordering of questions based among others on these metrics, will be explored in a near future.

3.2 Filtering annotated data

Once the selection and annotation phase described above is done, user-validated annotated sentences are stored in GRAMEX's database, together with their readability scores. Neural text analysis modules by their statistical nature may produce erroneous annotations (e.g., wrong morphological features). This is especially true since our NLP pipeline applies *pre-trained* dependency parsers to potentially out-of-domain data.

In order to detect annotations which are likely to be erroneous before exercise generation, a statistical filtering is applied. In brief, we compiled a corpus C_{err} of annotation errors by comparing our pipeline's annotations with a gold-standard (i.e., manually annotated) dataset made of 23,750 sentences coming from the French section of the Universal Dependency corpus (Nivre, 2016). From this corpus C_{err}, we experimented with various machine (deep and non-deep) learning algorithms in order to predict whether a given annotated sentence should be flagged as invalid. The best results were obtained by using gradient boosting (Friedman, 2002) reaching an F-score of 0.63.[11]

Note that, whatever the result of this filtering step is, the annotated sentence is kept in the database so that users can manually inspect or edit it should they want to.

3.3 Aligning annotated data with target grammatical phenomena

In order to control exercise generation with respect to target grammatical concepts,[12] we define an alignment between these and morpho-syntactic annotations generated by our NLP pipeline. This candidate selection is based on curricula-based predefined filters.

To facilitate the maintenance and extension of GRAMEX, these alignment filters are defined in configuration files and use a custom description language inspired by the Grew corpus query language (Guillaume, 2021) to specify which morpho-syntactic annotations contribute to a given target grammar concept. As an illustration, let us consider the following specification:

```
[upos=VERB&Mood=Ind&Tense=Fut]
```

Here, we specify a combination of annotations which are characteristics of sentences having a verb in future tense. It reads as follows: the sentence must contain a token whose part-of-speech tag is `VERB`, and whose morpho-syntactic features include `Mood,Ind` and `Tense,Fut` as key,value pairs.

Concretely, once a text is fully annotated by our NLP pipeline, these alignment filters are used to check the presence of any target grammatical concept in annotated sentences and, in case of success to keep their locations in the sentence (and store them together with the morpho-syntactic annotations in GRAMEX's database).

It is worth noting that, although most of the grammatical phenomena listed in official curricula are correctly flagged, some (such as simple past in French) are consistently not. We suspect this is due to the under-representation of these phenomena in parsers' training data. In order to circumvent this issue, we use a rule-based approach (e.g. a verb conjugation algorithm) to overwrite the annotations given by the parser. In case of ambiguity (same morpheme for several tenses), we keep all possible annotations in the database.

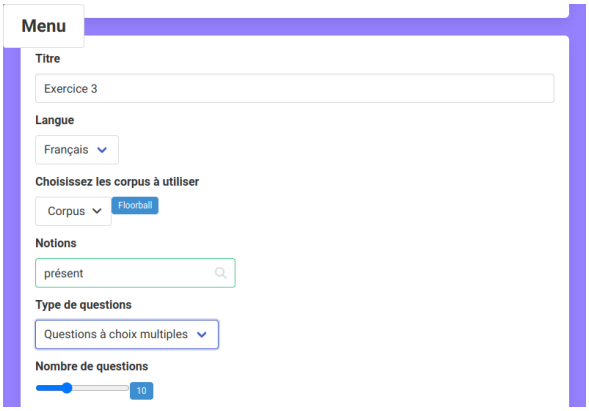

Figure 2: Teacher's interface to create exercises

3.4 Generating exercises

In order to create an exercise, users have to choose (i) a text (within the corpus of texts they have previously asked GRAMEX to annotate), (ii) a target grammatical concept to work on, (iii) a type of exercise (among the 4 types currently supported by GRAMEX, namely FiB, MtW, SC and MC), and (iv) a number of questions, see Figure 2. Exercises are also given a title and optional keywords to facilitate their indexing and reuse.

From this configuration, GRAMEX retrieves in the selected annotated text, the expected number of sentences exhibiting the target grammatical

[11] Filtering is work in progress, especially since all annotation errors are not of equal importance in our context.

[12] Recall that these concepts come from official curricula.

concept. A transformation rule is then applied to the corresponding sentences to turn them into the required exercise type. Note that, if the selected text contains more occurrences of the target grammatical concept than the required number of questions, a random selection is done. This is subject to modification in the future (see Section 5).

The user is then presented with the generated exercise, and has the option to replace questions and / or re-order them. Figure 3 shows an example of a FiB exercise on past perfect in French generated by GRAMEX.

3.5 Exporting exercises

Export refers to the possibility for users to download exercises in a given format. This is useful for creating backups, post-editing exercises or else sharing exercises with other teachers (who will import them). Supported export formats include JSON for programmatic uses, word documents for paper-based activities, and H5P components for use in dedicated (on-line or desktop) environments equipped with an H5P player (e.g., Lumi[13]).

Figure 4 gives an example of FiB and MC exercises exported in H5P format. Note that if needed, H5P components can be modified using the free H5P editor.[14]

3.6 Sharing exercises

Sharing refers to the possibility for teachers to give access to their exercises to other users. Sharing can either be public (that is, to all registered users) or else restricted to specific users only. Public exercises can be retrieved using a text-based search on their title, keywords and content.

Exercises shared with specific users (learners or groups of learners) can be accessed on invitation or else by using auto-generated access codes.

3.7 Taking exercises

Learners can access exercises from their dashboard directly if they have been invited by their teacher, or else by using their access code. In both cases, questions can be answered in a dedicated interface (see Figure 5). Once the exercise is completed, students are presented with a summary of their successes and failures (see Figure 6). All attempts can also be monitored by the teacher.

[13] https://lumi.education/en/

[14] https://h5p.org/installation

4 Implementation

GRAMEX relies on a client-server architecture, with a front-end in JavaScript / VueJS[15] and a back-end in Python. These components interact with a MySQL database following a classical Model-View-Controller design pattern (Krasner and Pope, 1988) as illustrated in Figure 7.

GRAMEX's database basically contains information about users (teachers and learners) and learning materials. These pieces of information are organized as follows. Teachers can manage learners' accounts, corpora (collections of annotated texts) and learning activities. Activities can either be a so-called *lessons* gathering textbooks and exercises, or *tests* (standalone exercises). Both lessons and tests can be shared with specific learners depending on their profile and / or teachers' pedagogical choices.

Note that GRAMEX's exercise generation module is used in a similar way when creating lessons or tests. The only difference lies in whether they are used in the context of formative or summative assessment (Sadler, 1998) by teachers (unlike exercises belonging to lessons, exercises from tests can be taken only once).

GRAMEX comes with a web user interface built with the Bulma CSS framework[16]. Users can use GRAMEX through responsive web pages designed for computers and tablets.

The back-end hosts GRAMEX's NLP pipeline and database. It also offers a REST API developed in Flask[17] allowing programs (including the front-end) to interact with the hosted components via the controller module. The back-end handles all data manipulations, from text annotation to exercise generation and export.

Note that the back-end also includes a typescript module responsible for generating H5P components, which are served by the API for download and reuse in other applications.

5 Current status

GRAMEX is under active development. Design choices are subject to modification depending on feedback from teachers. In the following subsections, we briefly report on a first 2-week experiment, which highlighted some limitations calling for further development.

[15] https://vuejs.org/

[16] https://bulma.io/

[17] https://flask.palletsprojects.com/

Numéro	Phrase	Réponse	Lisibilité	Actions
1	Il y avait une famille qui les _____ (avoir) _____ (voir) et qui s'est dit : «	avait ; vus	108	
2	De son côté, le chat était très content : il _____ (avoir) _____ (trouver) une chatte.	avait ; trouvé	105	
3	Et ça y est, elle l' _____ (avoir) _____ (choisir) !	avait ; choisi	104	
4	Mimi lui dit qu'il _____ (être) _____ (partir) à la recherche de son frère et ils rentrèrent à la maison.	était ; parti	93	
5	Pourtant, chaque fois, il _____ (avoir) _____ (essayer) de se retenir très fort.	avait ; essayé	92	

Figure 3: Fill-in-the-Blanks exercise generated targeting the past perfect tense

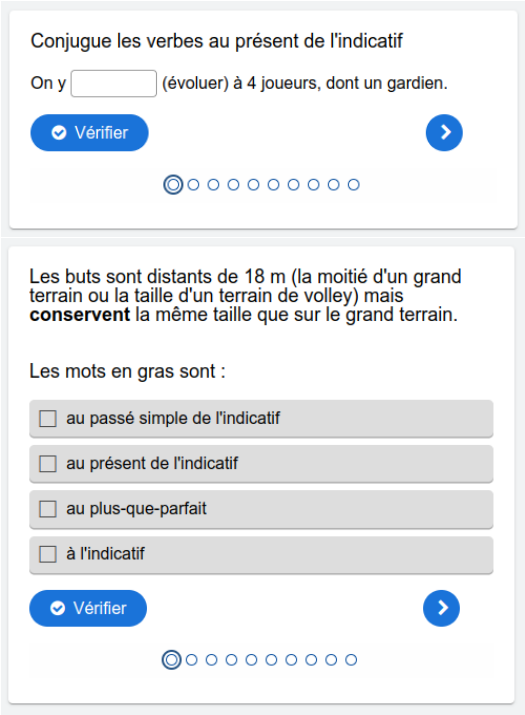

Figure 4: Exercises imported on an external H5P player

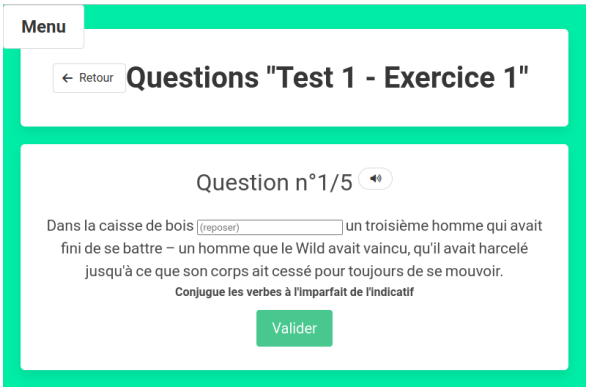

Figure 5: Learner's interface for taking questions

5.1 Report on experimental uses

GRAMEX has been used by a pool of 4 school teachers whose pupils' age range from 9 to 15 years old. They focused on the development of exercises from raw data (that is, they did not create lessons). In a few cases (where computers were available in classrooms), generated exercises were presented to pupils. In the end of the experiment, a questionnaire was sent to teachers to get their feedback on GRAMEX's usability (how easy / convenient it is to use GRAMEX?) and performance (how pertinent are the generated exercises?).

On the usability side, some pupils had difficulty logging in with auto-generated passwords. Teachers recommended the use of QR-codes to provide learners with a connection link. Teachers had troubles understanding the logic behind exercises and lessons. On the performance side, teachers encountered issues with complex web pages (extracted texts were noisy), and were wishing one could feed PDF files to the application. Finally teachers indicated they would need more control on sentence selection. For instance, they would like to be able to control the presence of various syntactic constructions in selected questions.

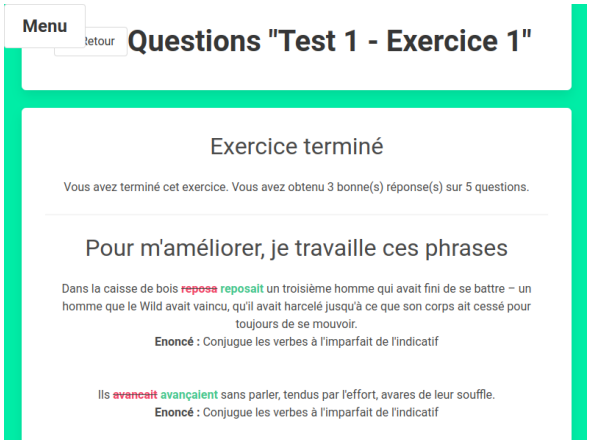

Figure 6: Learner's interface on exercise completion

5.2 Limitations and current work

As mentioned above, GRAMEX has been designed to facilitate exercise generation by providing users with a semi-automatic process requiring only a lightweight configuration. Candidate grammatical concepts (and corresponding morpho-syntactic annotations) are predefined and can be used out-of-the-box. It turns out that this configuration is not sufficient as teachers cannot precisely control the structure of generated questions. GRAMEX's workflow is thus being extended to give teachers the possibility to define syntactic constraints on the selected sentences. These constraints are written in the same description language as curricula-related filters (see Section 3.3).

Another main limitation of GRAMEX lies in its use of pre-trained neural modules for text analysis. As mentioned above, these modules are applied on unknown texts (potentially out-of-domain). Even though a statistical filter is applied, teachers cannot be guaranteed that the provided annotations (and thus exercises) are correct. We are currently working on the development of another annotation error detection module. Two paths are being considered : using ensemble techniques which would basically compare between annotations computed by distinct parsers following work by Surdeanu and Manning (2010), and using a rule-based approach were predicted dependency rules would be compared with a dependency grammar extracted from manually annotated data following work by Rehbein and Ruppenhofer (2018).

Another limitation of GRAMEX corresponds to the limited types of exercises it supports. This combined with the fact that FiB does not support answers which would deviate from original texts

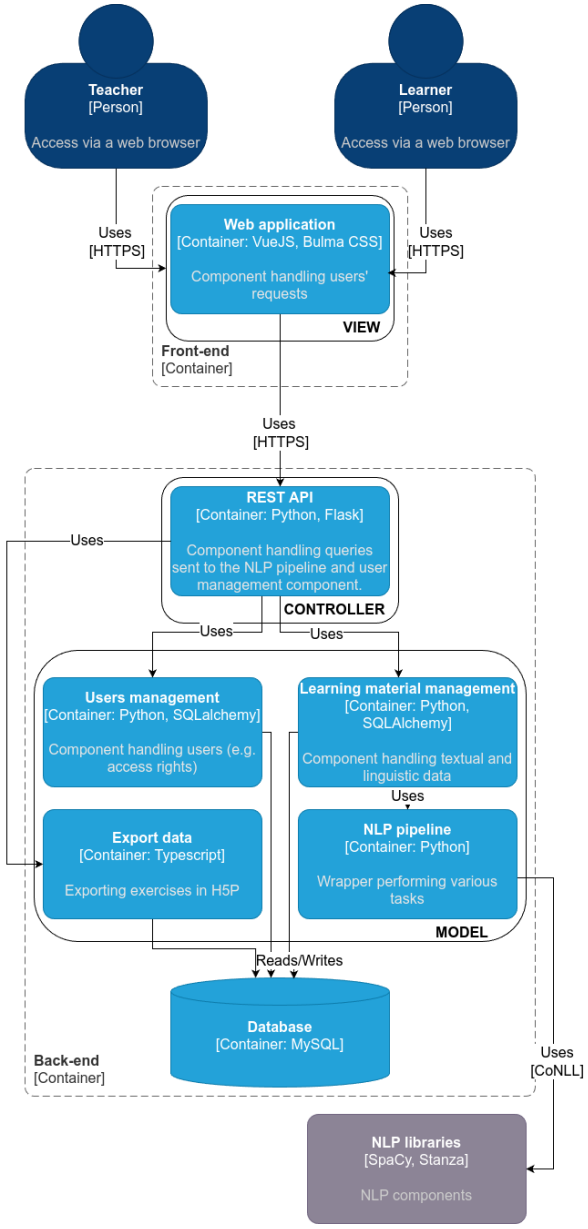

Figure 7: GRAMEX architecture

while being correct, makes it crucial to extend GRAMEX with new exercise types.

6 Conclusion and perspectives

We presented GRAMEX, an environment for CALL using state-of-the-art parsers to generate grammar exercises in line with official curricula. GRAMEX aims to help teachers to create adequate learning materials with minimal efforts. GRAMEX is work in progress and benefits from cooperations with field teachers. Future improvements include the configuration of exercises by means of an expressive search engine following Heck and Meurers (2022a), and the extension to new languages.

References

Itziar Aldabe, Maddalen Lopez de Lacalle, Montse Maritxalar, Edurne Martinez, and Larraitz Uria. 2006. Arikiturri: An automatic question generator based on corpora and nlp techniques. In *Intelligent Tutoring Systems*, pages 584–594, Berlin, Heidelberg. Springer Berlin Heidelberg.

Georges Antoniadis, Sandra Echinard, Olivier Kraif, Thomas Lebarbé, and Claude Ponton. 2006. Modélisation de l'intégration de ressources TAL pour l'apprentissage des langues : la plateforme MIRTO. *ALSIC - Apprentissage des Langues et Systèmes d'Information et de Communication*, 08(1):65–79.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

Clotilde Castagné-Véziès. 2018. La Grammaire à l'épreuve de la langue et de la métalangue. *Cognition, représentation, language (Corela)*, 16(1).

Thomas François and Cédrick Fairon. 2012. An "AI readability" formula for French as a foreign language. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 466–477, Jeju Island, Korea. Association for Computational Linguistics.

Jerome H. Friedman. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378. Nonlinear Methods and Data Mining.

Bruno Guillaume. 2021. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Kiev/Online, Ukraine.

Tanja Heck and Detmar Meurers. 2022a. Generating and authoring high-variability exercises from authentic texts. In *Proceedings of the 11th Workshop on NLP for Computer Assisted Language Learning*, pages 61–71, Louvain-la-Neuve, Belgium. LiU Electronic Press.

Tanja Heck and Detmar Meurers. 2022b. Parametrizable exercise generation from authentic texts: Effectively targeting the language means on the curriculum. In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 154–166, Seattle, Washington. Association for Computational Linguistics.

Nicolas Hernandez, Nabil Oulbaz, and Tristan Faine. 2022. Open corpora and toolkit for assessing text readability in French. In *Proceedings of the 2nd Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI) within the 13th Language Resources and Evaluation Conference*, pages 54–61, Marseille, France. European Language Resources Association.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Ayako Hoshino and Hiroshi Nakagawa. 2007. A cloze test authoring system and its automation. In *Proceedings of the 6th International Conference on Advances in Web Based Learning*, ICWL'07, page 252–263, Berlin, Heidelberg. Springer-Verlag.

María Ignacia-Dorronzoro and Estela Klett. 2007. Le rôle de la grammaire dans l'enseignement de la lecture en langue étrangère. *Éla. Études de linguistique appliquée*, 148:499–511.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Glenn E. Krasner and Stephen T. Pope. 1988. A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *Journal of Object Oriented Programming*, 1(3):26–49.

Nitin Madnani, Jill Burstein, John Sabatini, Kietha Biggers, and Slava Andreyev. 2016. Language muse: Automated linguistic activity generation for English language learners. In *Proceedings of ACL-2016 System Demonstrations*, pages 79–84, Berlin, Germany. Association for Computational Linguistics.

Ministère de l'Éducation Nationale de la Jeunesse et des Sports MENJS. 2018. Programmes d'enseignement. cycle des apprentissages fondamentaux (cycle 2), cycle de consolidation (cycle 3) et cycle des approfondissements (cycle 4) : modification (publication du bulletin officiel no mene1820169a).

Detmar Meurers, Ramon Ziai, Luiz Amaral, Adriane Boyd, Aleksandar Dimitrov, Vanessa Metcalf, and Niels Ott. 2010. Enhancing authentic web pages for language learners. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 10–18, Los Angeles, California. Association for Computational Linguistics.

Duy Van Ngo and Yannick Parmentier. 2023. Towards Sentence-level Text Readability Assessment for French. In *Second Workshop on*

Text Simplification, Accessibility and Readability (TSAR@RANLP2023), Varna, Bulgaria.

Joakim Nivre. 2016. Universal Dependencies: A cross-linguistic perspective on grammar and lexicon. In *Proceedings of the Workshop on Grammar and Lexicon: interactions and interfaces (GramLex)*, pages 38–40, Osaka, Japan. The COLING 2016 Organizing Committee.

Matthew Peacock. 1997. The effect of authentic materials on the motivation of EFL learners. *ELT Journal*, 51(2):144–156.

Naiara Perez and Montse Cuadros. 2017. Multilingual CALL framework for automatic language exercise generation from free text. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 49–52, Valencia, Spain. Association for Computational Linguistics.

Laura Perez-Beltrachini, Claire Gardent, and German Kruszewski. 2012. Generating grammar exercises. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 147–156, Montréal, Canada. Association for Computational Linguistics.

Ildikó Pilán, Elena Volodina, and Lars Borin. 2016. Candidate sentence selection for language learning exercises: from a comprehensive framework to an empirical evaluation. *Traitement Automatique des Langues*, 57(3):67–91.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Ines Rehbein and Josef Ruppenhofer. 2018. Sprucing up the trees – error detection in treebanks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 107–118, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

D. Royce Sadler. 1998. Formative assessment: revisiting the territory. *Assessment in Education: Principles, Policy & Practice*, 5(1):77–84.

Mihai Surdeanu and Christopher D. Manning. 2010. Ensemble models for dependency parsing: Cheap and good? In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 649–652, Los Angeles, California. Association for Computational Linguistics.

François Vincent. 2016. Articulation entre la grammaire, l'écriture et la lecture : résultats d'une recension d'écrits.

Elena Volodina, Ildikó Pilán, Lars Borin, and Therese Lindström Tiedemann. 2014. A flexible language learning platform based on language resources and web services. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3973–3978, Reykjavik, Iceland. European Language Resources Association (ELRA).